

1/5/1 (Item 1 from file: 351)
DIALOG(R) File 351:Derwent WPI
(c), 2000 Derwent Info Ltd. All rts. reserv.

012677455 **Image available**

WPI Acc No: 1999-483562
XRPX Acc No: N99-3605S

Monitoring arrangement for computer network with agents - has individual monitoring units for entire system, node and agent, that are driven sequentially

Patent Assignee: TOSHIBA KK (TOKE)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 11203140	A	19990730	JP 983234	A	19980109	199941 B

Priority Applications (No Type Date): JP 983234 A 19980109

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
JP 11203140	A	21	G06F-009/44	

Abstract (Basic): JP 11203140 A

NOVELTY - The arrangement has a system monitor (111), an agent monitor (115) and node monitor (113) to monitor the entire system, designated agent and designated node respectively. The monitors are operated sequentially. DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following: the monitoring method of the network; a debug system; debug method

USE - For monitoring computer network with agents for safety.

ADVANTAGE - The behavior of agent system is understood easily, with increased flexibility, smoothness and safety. The control operation is also improved. DESCRIPTION OF DRAWING(S) - The figure shows functional block diagram of the computer system. (111) System monitor; (113) Node monitor; (115) Agent monitor.

Dwg.1/15

Title Terms: MONITOR; ARRANGE; COMPUTER; NETWORK; AGENT; INDIVIDUAL; MONITOR; UNIT; SYSTEM; NODE; AGENT; DRIVE; SEQUENCE

Derwent Class: T01

International Patent Class (Main): G06F-009/44

International Patent Class (Additional): G06F-003/00; G06F-013/00; G06F-015/16

File Segment: EPI

1/5/2 (Item 1 from file: 347)

DIALOG(R) File 347:JAPIO

(c) 2000 JPO & JAPIO. All rts. reserv.

06261560 **Image available**

MONITOR DEVICE FOR AGENT SYSTEM, MONITOR METHOD, RECORDING MEDIUM RECORDING MONITORING PROGRAM, DEBUGGING DEVICE, DEBUG METHOD AND RECORDING MEDIUM RECORDING DEBUGGING PROGRAM

PUB. NO.: 11-203140 A]

PUBLISHED: July 30, 1999 (19990730)

INVENTOR(s): CHO KENTA

APPLICANT(s): TOSHIBA CORP

APPL. NO.: 10-003234 [JP 983234]

FILED: January 09, 1998 (19980109)

INTL CLASS: G06F-009/44; G06F-003/00; G06F-013/00; G06F-015/16

ABSTRACT

PROBLEM TO BE SOLVED: To grasp the conducts of an agent system by making a monitor monitor in units such as the whole, a node and an agent respectively.

SOLUTION: A system monitor 111 outputs what kinds of nodes and agents exist in an agent system and starts a node monitor 113 that monitors a designated node or an agent monitor 115 which monitors a designated agent. The monitor 113 outputs what kinds of agents exist in a corresponding node and starts the monitor 115 that monitors the designated agent. The monitor 115 outputs what node the corresponding agent exists in and what part of the current plan the agent is carrying out.

COPYRIGHT: (C)1999, JPO

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-203140

(43)公開日 平成11年(1999)7月30日

(51)Int.Cl.⁶
G 0 6 F 9/44
3/00
13/00
15/16

識別記号
5 5 2
6 5 2
3 5 5
4 3 0

F I
G 0 6 F 9/44
3/00
13/00
15/16

5 5 2
6 5 2 A
3 5 5
4 3 0 B

審査請求 未請求 請求項の数21 O.L (全 21 頁)

(21)出願番号 特願平10-3234

(22)出願日 平成10年(1998)1月9日

(71)出願人 000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者 長 健太
神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

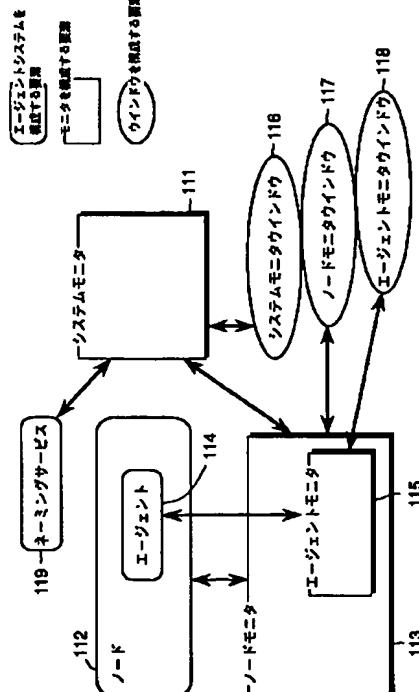
(74)代理人 弁理士 木内 光春

(54)【発明の名称】 エージェントシステムのモニタ装置、モニタ方法、モニタ用プログラムを記録した記録媒体、デバッグ装置、デバッグ方法及びデバッグ用プログラムを記録した記録媒体

(57)【要約】

【課題】 全体、ノード、エージェントといった単位でそれぞれモニタで監視することによってエージェントシステムの振る舞いを把握できるようにする。

【解決手段】 システムモニタ 111は、エージェントシステム内にどのようなノード及びエージェントが存在するかを出力し、指定されたノードを監視するノードモニタ 113又は指定されたエージェントを監視するエージェントモニタ 115を起動する。ノードモニタ 113は、対応するノードにどのようなエージェントが存在するかを出力し、指定されたエージェントを監視するエージェントモニタ 115を起動する。エージェントモニタ 115は、対応するエージェントがどのノードにいるか、及び当該エージェントが現在プランのどの部分を実行しているかを出力する。



【特許請求の範囲】

【請求項1】 エージェントが複数のノード上を移動しながら情報処理を行うエージェントシステムをモニタするためのモニタ装置において、

システム全体を監視するシステムモニタと、ノードを監視するノードモニタと、エージェントを監視するエージェントモニタと、を備えたことを特徴とするエージェントシステムのモニタ装置。

【請求項2】 前記各モニタから他のモニタを起動するように構成されたことを特徴とする請求項1記載のエージェントシステムのモニタ装置。

【請求項3】 前記システムモニタは、エージェントシステム内にどのようなノード及びエージェントが存在するかを出力し、指定されたノードを監視するノードモニタ又は指定されたエージェントを監視するエージェントモニタのうち少なくとも一方を起動するように構成されたことを特徴とする請求項1又は2記載のエージェントシステムのモニタ装置。

【請求項4】 前記ノードモニタは、対応するノードにどのようなエージェントが存在するかを出力し、指定されたエージェントを監視するエージェントモニタを起動するように構成されたことを特徴とする請求項1乃至3のいずれか一に記載のエージェントシステムのモニタ装置。

【請求項5】 前記ノードモニタは、対応するノードに、エージェントの動作を記録したログファイルとしてどのようなものが存在するかを出力し、指定されたログファイルに基づいてエージェントの動作を確認するように構成されたことを特徴とする請求項1乃至4のいずれか一に記載のエージェントシステムのモニタ装置。

【請求項6】 前記ノードモニタは、対応するノード上に存在するエージェントを監視するエージェントモニタを保持し、前記エージェントが他のノードへ移動すると前記エージェントモニタを破棄し、エージェントの移動先であるノードを監視しているノードモニタは、移動してきたエージェントを監視するエージェントモニタを生成して保持するように構成されたことを特徴とする請求項1乃至5のいずれか一に記載のエージェントシステムのモニタ装置。

【請求項7】 前記エージェントモニタは、対応するエージェントがどのノードにいるか、及び当該エージェントが現在プランのどの部分を実行しているかを出力するように構成されたことを特徴とする請求項1乃至6のいずれか一に記載のエージェントシステムのモニタ装置。

【請求項8】 前記エージェントモニタは、対応するエ

ージェントのプランのうち所望の部分にブレークポイントを設定するように構成されたことを特徴とする請求項1乃至7のいずれか一に記載のエージェントシステムのモニタ装置。

【請求項9】 前記エージェントモニタは、対応するエージェントのプランのうち所望の部分に、エージェントの動作についてログの記録を制御するためのロギングポイントを設定するように構成されたことを特徴とする請求項1乃至8のいずれか一に記載のエージェントシステムのモニタ装置。

【請求項10】 エージェントが複数のノード上を移動しながら情報処理を行うエージェントシステムをモニタするためのモニタ方法において、システム全体を監視するプロセスと、ノードを監視するプロセスと、エージェントを監視するプロセスと、を用いることを特徴とするエージェントシステムのモニタ方法。

【請求項11】 前記システム全体を監視するプロセスは、

エージェントシステム内にどのようなノード及びエージェントが存在するかを出力するステップと、ノード又はエージェントのうち少なくとも一方を指定するためのステップと、指定されたノードを監視するプロセス又は指定されたエージェントを監視するプロセスのうち少なくとも一方を起動するステップと、を実行することを特徴とする請求項10記載のエージェントシステムのモニタ方法。

【請求項12】 前記ノードを監視するプロセスは、対応するノードにどのようなエージェントが存在するかを出力するステップと、エージェントを指定するためのステップと、指定されたエージェントを監視するプロセスを起動するステップと、を実行することを特徴とする請求項10又は11記載のエージェントシステムのモニタ方法。

【請求項13】 前記エージェントを監視するプロセスは、対応するエージェントがどのノードにいるか、及び当該エージェントが現在プランのどの部分を実行しているかを出力するステップを実行することを特徴とする請求項10乃至12のいずれか一に記載のエージェントシステムのモニタ方法。

【請求項14】 エージェントが複数のノード上を移動しながら情報処理を行うエージェントシステムを、コンピュータを用いてモニタするためのモニタ用プログラムを記録した記録媒体において、当該モニタ用プログラムは前記コンピュータに、システム全体を監視させ、ノードを監視させ、

エージェントを監視させることを特徴とするエージェントシステムのモニタ用プログラムを記録した記録媒体。

【請求項15】 知識及びアクション定義に基づいてエージェントのプランを生成するプランナと、生成されたプランを実行することによってエージェントによる情報処理を実現する手段と、を有するエージェントシステムをデバッグするためのデバッグ装置において、
デバッグの対象とする知識、アクション定義及びプランナを指定する手段と、

指定された前記対象を用いて、与えられたゴールを達成するためのスクリプトを生成する手段と、
生成されたスクリプトを出力する手段と、
を有することを特徴とするエージェントシステムのデバッガ装置。

【請求項16】 生成された前記スクリプトに含まれるサブゴールを抽出する手段と、
指定されたサブゴールに対応するスクリプトを出力する手段と、
を有することを特徴とする請求項15記載のエージェントシステムのデバッガ装置。

【請求項17】 ゴールとサブゴール及び前記ゴールに対応するスクリプトと前記サブゴールに対応する部分的なスクリプトを、階層関係に基づいて表示するように構成されたことを特徴とする請求項15又は16記載のエージェントシステムのデバッガ装置。

【請求項18】 与えられたスクリプトの所望の部分にブレークポイントを設定する手段と、
与えられたスクリプトの実行に用いる所望の変数を確認の対象として指定する手段と、
与えられたスクリプトを指定された態様で実行する手段と、
設定された前記ブレークポイントにおいてスクリプトの実行を停止する手段と、
指定された変数に関する情報を出力する手段と、
を有することを特徴とする請求項15乃至17のいずれかに記載のエージェントシステムのデバッガ装置。

【請求項19】 知識及びアクション定義に基づいてエージェントのプランを生成するプランナと、生成されたプランを実行することによってエージェントによる情報処理を実現する手段と、を有するエージェントシステムをデバッグするためのデバッガ方法において、
デバッグの対象とする知識、アクション定義及びプランナを指定するステップと、
指定された前記対象を用いて、与えられたゴールを達成するためのスクリプトを生成するステップと、
生成されたスクリプトを出力するステップと、
を含むことを特徴とするエージェントシステムのデバッガ方法。

【請求項20】 与えられたスクリプトの所望の部分にブレークポイントを設定するステップと、

与えられたスクリプトの実行に用いる所望の変数を確認の対象として指定するステップと、

与えられたスクリプトを指定された態様で実行するステップと、

設定された前記ブレークポイントにおいてスクリプトの実行を停止するステップと、

指定された変数に関する情報を出力するステップと、
を含むことを特徴とする請求項19記載のエージェントシステムのデバッガ方法。

10 【請求項21】 知識及びアクション定義に基づいてエージェントのプランを生成するプランナと、生成されたプランを実行することによってエージェントによる情報処理を実現する手段と、を有するエージェントシステムをコンピュータを用いてデバッガするためのデバッガ用プログラムを記録した記録媒体において、
当該デバッガ用プログラムは前記コンピュータに、
デバッガの対象とする知識、アクション定義及びプランナの指定を受け付けさせ、

20 指定された前記対象を用いて、与えられたゴールを達成するためのスクリプトを生成させ、
生成されたスクリプトを出力させることを特徴とするエージェントシステムのバグ用プログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ネットワーク上に分散して存在する情報をエージェントを用いて処理する技術の改良に関するもので、特に、エージェントによる動作の柔軟性と安全性を高めたものである。

【0002】

【従来の技術】 【エージェントシステム】 従来から、コンピュータのネットワーク上に分散した情報を処理する技術として、エージェントシステムが知られている。エージェントとは、ソフトウェア上の処理単位であり、周囲の状況に応じて自律的に動作するものである。エージェントシステムは、このようなエージェントが、ネットワークを構成するノード上を必要に応じて移動しながら情報収集などの処理を行うシステムである。ここで、ノードとは、ネットワークを構成する論理的な単位であり、一つのマシンすなわちコンピュータ上に複数存在し得る。

【0003】 図7は、このようなエージェントシステムの一例として、本出願人が特願平7-176181で提案しているエージェントシステムについて、その構成例を示す機能ブロック図である。この図に示すエージェントシステムは、複数のノード800をネットワーク800Nで接続したものであり、多数設けることができるノードのうち図7では2つだけを例示している。そして、ノード800のうち、ユーザがエージェント生成に使用するノードをローカルノード(800L)と呼び、生成

されたエージェントが移動して行く先のノードをリモートノード(800R)と呼ぶ。

【0004】このエージェントシステムにおいて、各ノード800は、ユーザがエージェントを生成する操作を行なったり、エージェントが情報処理を行なった結果を受け取るための入出力手段803(L, R)を有する。また、各ノードのエージェント管理手段804(L, R)は、エージェントを生成したり、役割を終えたエージェントを消去する他、エージェントの情報を他のノードへ転送することによって、エージェントを他のノードへ移動させたり、他のノードから同様に移動してくるエージェントを受け入れる手段である。ユーザは、このようなエージェントシステムを用いて何らかの情報処理を行ないたい場合、ローカルノード800Lにおいて、入出力手段803Lからエージェント管理手段804Lに指示を与えることによってエージェントを生成させる。

【0005】そして、最も基本的な例を示せば、生成されたエージェントに対して、ユーザが入出力手段803Lからスクリプトを与える。スクリプトは、エージェントの行動プログラムであり、どのノードへ移動し、どのような処理を行う、といった内容を具体的に記述したものである。スクリプトのより具体的な例としては、例えば、ノードAに移動してファイルaのコピーをユーザのノードに送信し、次にノードBに移動してファイルbのコピーをユーザのノードに送信し…といった内容が考えられる。そして、各ノードに備えられた解釈実行手段802(L, R)は、このようなスクリプトを実行することによってエージェントを活動させ、これによって目的とする情報処理を実現する。

【0006】この場合、各ノードに備えられたエージェント情報記憶手段801(L, R)が、エージェントに必要な情報を記憶する。エージェントに必要な情報は、例えば、前記のスクリプトのほか、スクリプトの解釈実行に必要な各変数(スクリプト変数と呼ばれる)や、必要な場合は、エージェントが収集した情報やファイルなどである。また、エージェントのスクリプトに記述される命令としては、一つのノード上だけで実行できる命令のほか、エージェントを他のノードへ移動させるための移動命令がある。解釈実行手段802Lは、スクリプトの命令を順次実行し、移動命令の実行が必要になった場合には、移動先のノードを指定して、エージェントの移動をエージェント管理手段804に依頼する。

【0007】このようなエージェントシステムでは、ユーザが、いくつかのファイルをネットワーク上から収集したいような場合、この目的を達成するための行動プログラムをエージェントに持たせてネットワーク上に送り出せばよく、送り出されたエージェントは、与えられたスクリプトに基づいて自律的に活動する。このため、ユーザのノードとエージェントとの間で通信を始終維持する必要はないことから、ftpやtelnetといった従来のネ

ットワーク機能と比べて回線障害に強いという利点がある。

【0008】【プランニングを用いる構成例】図7に示したエージェントシステムに対して、エージェントの行動プログラムであるスクリプトを、状況に応じて変化させることができるエージェントシステムも知られている。

【0009】すなわち、近年のようにネットワークが大規模化・複雑化し、特に、インターネットのような広域ネットワークと接続されることによっていわゆる開放型ネットワークになると、ファイルの所在などのようなネットワークの構成要素がしばしば変化するようになる。ところが、図7に示した上記のようなエージェントシステムでは、エージェントは生成される時点で固定されたスクリプトを与えられるため、状況に応じて行動を変更することができない。そこで、このような変化に柔軟に対応するため、人手を煩わせずにエージェントの行動を自動的に変化させる技術として、本出願人は、プランニング機能を持ったエージェントシステムを出願している。

【0010】この技術では、エージェントの行動プログラムはプランと呼び、プランを生成することをプランニングと呼ぶ。そして、この技術では、状況に応じてプランを適宜作り直すことによって、ネットワークの構成要素の変化に対応する。なお、ネットワークの構成などの変化に対応してプランニングを再度やり直すことを再プランニングと呼ぶ。

【0011】このようなエージェントシステムの構成例を図8の機能ブロック図に示す。この技術において、プランの生成に用いる情報としては、「知識」と呼ばれる情報とアクション定義とが挙げられる。このうち「知識」は、エージェントの動作、特にプランニングに利用する各種の情報であり、その一例として、どのファイルがどのノードに存在するかといったネットワークの構成要素に関する情報を含む。例えば図8の例では、このようなネットワークの構成に関する知識を、ローカル情報記憶手段1Lに保存しておき、ネットワークの構成に変化があったときは、更新手段2Lが、自動検出や手作業などによって、このような変化を知識に反映させる。また、アクション定義とは、プランを構成する部品として、どのような種類の命令(アクション)が使えるかを表す情報であり、エージェント情報記憶手段3に格納しておく。

【0012】このようなエージェントシステムでは、エージェントの生成を指示するユーザは、達成したいゴールをスクリプトの代わりにノードに与える。ここで、ゴールとは、情報処理の目的として達成したい状態を予め定められた文法で記述したものである。すると、プラン生成手段5は、与えられた知識を参照しながら、アクション定義に含まれる各種のアクションを組み合わせるこ

とによって、ゴールを達成するためのプランを生成する。このようなエージェントシステムでは、ネットワークの構成の変化は、プランニングや再プランニングの際に、知識を介してエージェントのプランに反映されるので、エージェントは、人手を介さずに状況の変化に対応し、柔軟に行動を変化させることができる。

【0013】なお、このようなプランを生成する手段は、「プランナ」とも呼ばれ、その実体はプランニングの手続きを表すプログラムの一種である。また、エージェントの行動プログラムやその各部分を呼ぶ広義の概念がスクリプトであり、プランというときは、特に、図8に示したようなプランニングを行うエージェントによって生成されたスクリプトの全体を指す。

【0014】【プランニングの例】続いて、上記のようなプランニングを用いたエージェントシステムの具体的な動作手順を図9に例示する。この手順では、ユーザが、情報処理のゴールとしてエージェントに対する要求の記述（要求記述）を入力すると（ステップ201）、必要な初期化が行なわれた後（ステップ202）、プランが生成される（ステップ203）。なお、処理は、ゴールが既に達成されているなど終了条件の判定結果に応じて終了する（ステップ204、205）。

【0015】すなわち、このような終了条件が満たされるまでは、ゴールを達成するために実行を要するプランの実行が行われる（ステップ204）。プランの実行では、プランに含まれる各命令を順次実行し、実行する命令が移動命令の場合にはエージェントをノード間で移動させる処理（goアクションと呼ばれる）が実行される（ステップ206～208）。また、各命令の実行やgoアクションの実行に失敗した場合は、必要に応じて新たなプランを生成する。

【0016】ここで、プランの生成に用いるアクション定義には、アクション（動作）の種類ごとに、事前条件と事後条件とが定義されている。このうち事前条件とは、どんな条件が満たされていればその動作を実行できるかを表し、事後条件は、その動作を実行するとどのような条件が作り出されるかを表す。例えば、「ファイルをコピーする」という動作を行うには、「現在いるノードにファイルが存在する」という事前条件が必要であり、コピーの動作を行なった結果として「ファイルのコピーが存在する」という事後条件が産み出される。

【0017】プランの生成は、最終的なゴールを事後条件として産み出す動作を発見し、この動作の事前条件を事後条件として産み出すさらに別の動作を発見する、という処理を続けることによって、プランを実行する前の状態（現在の状態）と最終的なゴールとの間をつなぐ動作の列を得ることである。なお、図10は、生成途中におけるプランの例を示す図であり、この例では、動作P2の一方の事前条件C5と、動作P3の事前条件C7について、これら事前条件を事後条件として産み出す動作

がまだ見つかっていない。このように、事後条件として産み出す他の動作がまだ見つかっていない事前条件は未達成ゴールと呼ばれる。

【0018】このようなプラン生成の処理は、ゴールの側から因果を逆に遡って行ない、プランの実行を開始する時点で存在している状態（現在の状態）に到達すると終了する。図11は、このような処理によって完成したプランの例を示す図である。

【0019】続いて、プラン生成の具体的な手順を図11に示す。すなわち、この手順では、ゴールを記録しておくゴールリストの一部を、図10に示したような未達成ゴールを記録しておく未達成ゴールリストとしており、次のような処理を行う。まず、ゴールリストに未達成ゴールが存在しなくなるまで（ステップ401）、未達成ゴールリストから未達成ゴールを1つずつ選択し（ステップ402）、ゴールが満足されている場合を除いて（ステップ403）、次のような動作を行う。すなわち、ゴールである事前条件を事後条件によって達成可能な動作が存在すれば（ステップ404）この動作を選択し（ステップ405）、このように選択した動作（選択動作）を図11に示したような動作の系列（プラン木）に追加する（ステップ406）。

【0020】また、ゴールを達成可能な動作が存在しない場合は、ゴールが不確実知識で達成可能かを判断する。ここで、不確実知識とは、ネットワークの構成に関する知識のうち、他のノードで実際に何らかの処理を行なってみないと具体的な値がわからない知識である。ゴールが不確実知識で達成可能な場合はこの不確実知識を選択動作としてプラン木に追加するが（ステップ405）、不確実知識でも達成不可能な場合は、処理をバックトラックさせ（ステップ408）、現在の未達成ゴールを生じさせている動作を他の動作に置き換えて再度処理を行う。

【0021】例えば、ユーザが使用するノードのノード知識で、「ファイルaがノードAに存在する」とされているとする。この場合、ファイルaを得るというゴールをユーザが与えると、ノードAに存在するという知識が参照されるので、生成されたエージェントのプランは、「ノードAに移動してファイルaのコピーをユーザのノードに送信する」といった内容になる。

【0022】しかし、エージェントがノードAに移動した時点で、ファイルaはノードBに移動されていると、ファイルaが発見できないためにプランは実行失敗となり、ノードA上で再プランニングが行なわれる。このとき、ノードBのノード知識がファイルの移動にあわせて更新されており、「ファイルaはノードBに存在する」と変更されている場合、新しいプランは「ノードBに移動してファイルaのコピーをユーザのノードに送信する」という内容に変更される。この結果、エージェントはノードBに自律的に移動するし、ファイルaを無事

発見してユーザのノードに送信することができる。

【0023】また、このように生成されたプランに基づいて、エージェントがノード間で移動する場合の手順を図13に示す。この例では、ユーザがエージェントを生成したローカルノードから、他のノードであるリモートノードへエージェントが移動するものとする。この場合、ローカルノードからの移動要求（ステップ501）を受信したリモートノードは（ステップ502）、エージェント用のプロセスを設定する（ステップ503）。

【0024】続いて、リモートノードから、プロセスの設定が完了した旨の通知（ステップ504）を受信したローカルノードは（ステップ505）、エージェントのプランや変数領域などのエージェント情報をリモートノードに送信する（ステップ506）。このエージェント情報を受信したリモートノードは（ステップ507）、エージェント情報を格納し（ステップ508）、ローカルノードへ移動成功の通知を送信し（ステップ509）、プランの解釈実行を開始する（ステップ510）。一方、成功の通知を受信したローカルノードは（ステップ511）、不要になったエージェント用のプロセスを消去する（ステップ512）。

【0025】〔エージェントのライフサイクル〕次に、以上のようにプランニングを行うエージェントのライフサイクルを示す概念図を図14に示す。すなわち、エージェントは、ゴール投入と共に生成されて活動を開始すると、まず、プランを生成するプランニングフェーズPから開始し、生成されたプランにしたがい、プランを実行する実行フェーズEやノード間で移動する移動フェーズMに移行し、実行や移動の失敗に応じてこれらの各フェーズ間を移行しながら活動する。そして、当初与えられたゴールを達成すれば正常終了となり、ゴールが達成できずにプランニング自体にも失敗すると完全失敗となって終了する。

【0026】〔ファームを用いる例〕また、1つのノード上に、ファームと呼ばれる領域をいくつか設けたエージェントシステムも考えられる。ここで、ファームとは、情報処理の目的や分野ごとに設定されるエージェントの活動領域であり、フィールドとも呼ばれる。このようなファーム（フィールド）は、一つのノード上に複数存在することができ、メモリなどの資源やプランの生成・実行に用いる情報は、ファーム（フィールド）毎に設けられる。

【0027】図15は、ネットワークNに複数のホストH（マシン）が接続され、各ホストH上には1つずつのノードXが存在し、ノードX上に複数のフィールドFLが存在する例を示す概念図である。このようなエージェントシステムでは、プラン生成に用いる知識がファーム（フィールドFL）ごとに分けられており、これによって、エージェントはプランニングなどに必要な情報を検索する際、余分な情報まで参照する必要がないので、情

報処理が効率化される。なお、プラン生成に用いられる知識はその知識を所持している主体によって、ファームが所持しているファーム知識、エージェントが所持しているエージェント知識などに分けることができる。

【0028】

【発明が解決しようとする課題】しかしながら、上記のような従来技術には、次のような問題点が存在していた。

（1）まず、上記のように、エージェントシステムは、複数のノードを有するだけでなく、これらノード上を複数のエージェントが動き回るという複雑なシステムである。そして、従来、ネットワークに含まれる個々のノード毎に、状態を表示するモニタは知られていた。しかし、エージェントシステム全体について、どのようなノード及びエージェントが存在しているかを管理したり、特定のノードに存在するエージェントを把握したり、特定のエージェントがノード上を移動していく様子を追跡するモニタは知られていなかった。このため、システム全体の状態やエージェントの移動経路を把握してシステム全体の管理を効果的に行なったり、ノードやエージェントが持っている情報をデバッグするなどの作業が困難であった。

【0029】（2）また、エージェントシステムは、ゴールが与えられると、知識やアクション定義に基づいて、ゴールを達成するためのプランを生成する。しかし、従来は、与えられたゴールに対して正しいスクリプトが生成されるかどうかを確認するツールが存在しなかった。このため、知識やアクション定義のデバッグが困難であった。

【0030】また、与えられたゴールは、通常、1つ又は複数のサブゴールを派生させる。このサブゴールとは、最終的なゴールを達成するために、プラン生成の際に中間的に作成されるゴールである。例えば、最終的な1つのゴールを達成するために、2つの別々の必須条件が必要な場合、それぞれの条件がサブゴールとなり、各サブゴールを達成するためにはそのための部分的なスクリプトが生成される。

【0031】このため、最終的なゴールに対応して生成されたスクリプト全体について、システム内からテキストファイルを取り出せても、ゴールとサブゴールの関係や、サブゴールを達成するためにはどのようなスクリプトが用いられるかを理解することは困難であり、所望の部分について適切なデバッグを行うことが困難であった。

【0032】（3）また、従来のスクリプトは記述された命令を単に順番に実行するだけのものであったため、その動作を把握することも比較的容易であった。しかし、スクリプトが複雑化すると、その動作の把握は困難となる。

【0033】すなわち、本出願人は、どのような条件の

時にどのような命令が使用できるかを表すアクション定義と、ある命令の動作として具体的にはどのような動作を行うかというスクリプトの実体とを、別々に記述できるエージェントシステムを出願している。このエージェントシステムでは、スクリプトの実体を、アクション定義の記述文法に拘束されずに自由に記述できる。このため、各種制御構文や変数を用いることによって、スクリプト中に複雑な動作を記述可能となる。

【0034】このように複雑なスクリプトをデバッグしようとする場合は、スクリプトのどの部分がどのような順序で実行され、また、それに伴ってどの変数がどのように変化するかを把握することが不可欠となる。しかし、従来は、このように複雑なスクリプトをデバッグするための技術は存在しなかったので、複雑なスクリプトのデバッグは特に困難であった。

【0035】本発明は、上記のような従来技術の問題点を解決するために提案されたもので、その目的は、エージェントシステムの管理と開発とを効果的に支援する技術を提供することである。より具体的には、本発明の目的は次のとおりである。

(1) 全体、ノード、エージェントといった単位でそれぞれモニタで監視することによってエージェントシステムの振る舞いを把握できること。

(2) エージェントシステムのデバッグを効果的に支援すること。

(3) 複雑なスクリプトを効果的にデバッグする技術を提供すること。

【0036】

【課題を解決するための手段】上記の目的を達成するため、請求項1の発明は、エージェントが複数のノード上を移動しながら情報処理を行うエージェントシステムをモニタするためのモニタ装置において、システム全体を監視するシステムモニタと、ノードを監視するノードモニタと、エージェントを監視するエージェントモニタと、を備えたことを特徴とする。請求項10の発明は、請求項1の発明を方法の観点から把握したもので、エージェントが複数のノード上を移動しながら情報処理を行うエージェントシステムをモニタするためのモニタ方法において、システム全体を監視するプロセスと、ノードを監視するプロセスと、エージェントを監視するプロセスと、を用いることを特徴とする。請求項14の発明は、請求項1の発明をコンピュータプログラムを記録した記録媒体の観点から把握したもので、エージェントが複数のノード上を移動しながら情報処理を行うエージェントシステムを、コンピュータを用いてモニタするためのモニタ用プログラムを記録した記録媒体において、当該モニタ用プログラムは前記コンピュータに、システム全体を監視させ、ノードを監視させ、エージェントを監視させることを特徴とする。請求項1、10、14の発明では、システム全体、ノード及びエージェントをそれ

ぞれ専用のモニタによって監視できるので、エージェントシステムの全体と各部分の双方について状態を効果的に把握でき、エージェントシステムの管理やデバッグが容易になる。

【0037】請求項2の発明は、請求項1記載のエージェントシステムのモニタ装置において、前記各モニタから他のモニタを起動するように構成されたことを特徴とする。請求項3の発明は、請求項1又は2記載のエージェントシステムのモニタ装置において、前記システムモニタは、エージェントシステム内にどのようなノード及びエージェントが存在するかを出し、指定されたノードを監視するノードモニタ又は指定されたエージェントを監視するエージェントモニタのうち少なくとも一方を起動するように構成されたことを特徴とする。請求項11の発明は、請求項3の発明を方法の観点から把握したもので、請求項10記載のエージェントシステムのモニタ方法において、前記システム全体を監視するプロセスは、エージェントシステム内にどのようなノード及びエージェントが存在するかを出力するステップと、ノード

10 又はエージェントのうち少なくとも一方を指定するためのステップと、指定されたノードを監視するプロセス又は指定されたエージェントを監視するプロセスのうち少なくとも一方を起動するステップと、を実行することを特徴とする。請求項4の発明は、請求項1乃至3のいずれか一に記載のエージェントシステムのモニタ装置において、前記ノードモニタは、対応するノードにどのようなエージェントが存在するかを出し、指定されたエージェントを監視するエージェントモニタを起動するよう構成されたことを特徴とする。請求項12の発明は、

20 請求項4の発明を方法の観点から把握したもので、請求項10又は11記載のエージェントシステムのモニタ方法において、前記ノードを監視するプロセスは、対応するノードにどのようなエージェントが存在するかを出力するステップと、エージェントを指定するためのステップと、指定されたエージェントを監視するプロセスを起動するステップと、を実行することを特徴とする。請求項2~4、11、12の発明では、各モニタから他の種類のモニタを起動できるので、エージェントシステムの状態を多面的に把握することが容易になる。

30 40 【0038】請求項5の発明は、請求項1乃至4のいずれか一に記載のエージェントシステムのモニタ装置において、前記ノードモニタは、対応するノードに、エージェントの動作を記録したログファイルとしてどのようなものが存在するかを出し、指定されたログファイルに基づいてエージェントの動作を確認するよう構成されたことを特徴とする。請求項5の発明では、所望のエージェントのログファイルに基づいてエージェントの動作を再現などによって確認できるので、個々のエージェントの動作を容易に検討することができる。

50 【0039】請求項6の発明は、請求項1乃至5のいず

れか一に記載のエージェントシステムのモニタ装置において、前記ノードモニタは、対応するノード上に存在するエージェントを監視するエージェントモニタを保持し、前記エージェントが他のノードへ移動すると前記エージェントモニタを破棄し、エージェントの移動先であるノードを監視しているノードモニタは、移動してきたエージェントを監視するエージェントモニタを生成して保持するように構成されたことを特徴とする。請求項6の発明では、エージェントがノード間で移動すると、移動したエージェントのエージェントモニタは、移動元ノードでは破棄され、移動先ノードで再度生成されるので、エージェントモニタをエージェントと共に送信する必要がなく、そのような送信に要する所要時間や手数が不要となる。

【0040】請求項7の発明は、請求項1乃至6のいずれか一に記載のエージェントシステムのモニタ装置において、前記エージェントモニタは、対応するエージェントがどのノードにいるか、及び当該エージェントが現在プランのどの部分を実行しているかを出力するように構成されたことを特徴とする。請求項13の発明は、請求項7の発明を方法の観点から把握したもので、請求項10乃至12のいずれか一に記載のエージェントシステムのモニタ方法において、前記エージェントを監視するプロセスは、対応するエージェントがどのノードにいるか、及び当該エージェントが現在プランのどの部分を実行しているかを出力するステップを実行することを特徴とする。請求項7、13の発明では、エージェントがどのノードにいて、プランのどの部分を実行しているかが出力できるので、所望のエージェントの状態把握が容易になる。

【0041】請求項8の発明は、請求項1乃至7のいずれか一に記載のエージェントシステムのモニタ装置において、前記エージェントモニタは、対応するエージェントのプランのうち所望の部分にブレークポイントを設定するように構成されたことを特徴とする。請求項8の発明では、エージェントモニタからエージェントにブレークポイントを設定することができるので、エージェントの状態の監視からデバッグへ容易に移行でき、作業が効率化される。

【0042】請求項9の発明は、請求項1乃至8のいずれか一に記載のエージェントシステムのモニタ装置において、前記エージェントモニタは、対応するエージェントのプランのうち所望の部分に、エージェントの動作についてログの記録を制御するためのロギングポイントを設定するように構成されたことを特徴とする。請求項9の発明では、エージェントのプランにロギングポイントを設定することによって、必要な部分だけのログを容易に記録することができる。

【0043】請求項15の発明は、知識及びアクション定義に基づいてエージェントのプランを生成するプラン

ナと、生成されたプランを実行することによってエージェントによる情報処理を実現する手段と、を有するエージェントシステムをデバッグするためのデバッグ装置において、デバッグの対象とする知識、アクション定義及びプランナを指定する手段と、指定された前記対象を用いて、与えられたゴールを達成するためのスクリプトを生成する手段と、生成されたスクリプトを出力する手段と、を有することを特徴とする。請求項19の発明は、請求項15の発明を方法の観点から把握したもので、知識及びアクション定義に基づいてエージェントのプランを生成するプランナと、生成されたプランを実行することによってエージェントによる情報処理を実現する手段と、を有するエージェントシステムをデバッグするためのデバッグ方法において、デバッグの対象とする知識、アクション定義及びプランナを指定するステップと、指定された前記対象を用いて、与えられたゴールを達成するためのスクリプトを生成するステップと、生成されたスクリプトを出力するステップと、を含むことを特徴とする。請求項21の発明は、請求項15の発明をコンピュータプログラムを記録した記録媒体の観点から把握したもので、知識及びアクション定義に基づいてエージェントのプランを生成するプランナと、生成されたプランを実行することによってエージェントによる情報処理を実現する手段と、を有するエージェントシステムをコンピュータを用いてデバッグするためのデバッグ用プログラムを記録した記録媒体において、当該デバッグ用プログラムは前記コンピュータに、デバッグの対象とする知識、アクション定義及びプランナの指定を受け付けさせ、指定された前記対象を用いて、与えられたゴールを達成するためのスクリプトを生成させ、生成されたスクリプトを出力させることを特徴とする。請求項15、19、21の発明では、指定されたデバッグ対象を用いて、与えられたゴールを達成するためのスクリプトが生成され出力されるので、ユーザは、所望のデバッグ対象について意図した通りのスクリプトが生成されているかどうかを容易に確認することができる。

【0044】請求項16の発明は、請求項15記載のエージェントシステムのデバッグ装置において、生成された前記スクリプトに含まれるサブゴールを抽出する手段と、指定されたサブゴールに対応するスクリプトを出力する手段と、を有することを特徴とする。請求項16の発明では、最終的なゴールに対応するスクリプト全体を表示して正しいかどうかを判断するだけでなく、所望のサブゴールに対応する部分的なスクリプトを表示して詳細に検討できるので、スクリプトの理解が容易になる。

【0045】請求項17の発明は、請求項15又は16記載のエージェントシステムのデバッグ装置において、ゴールとサブゴール及び前記ゴールに対応するスクリプトと前記サブゴールに対応する部分的なスクリプトを、階層関係に基づいて表示するように構成されたことを特

徴とする。請求項17の発明では、ゴールとサブゴール及びゴールに対応するスクリプトとサブゴールに対応する部分的なスクリプトについて、上位階層の表示と下位階層の表示を切り替えるなど階層関係に基づいて表示することによって、所望の範囲を容易に確認することができる。

【0046】請求項18の発明は、請求項15乃至17のいずれか一に記載のエージェントシステムのデバッグ装置において、与えられたスクリプトの所望の部分にブレークポイントを設定する手段と、与えられたスクリプトの実行に用いる所望の変数を確認の対象として指定する手段と、与えられたスクリプトを指定された態様で実行する手段と、設定された前記ブレークポイントにおいてスクリプトの実行を停止する手段と、指定された変数に関する情報を出力する手段と、を有することを特徴とする。請求項20の発明は、請求項18の発明を方法の観点から把握したもので、請求項19記載のエージェントシステムのデバッグ方法において、与えられたスクリプトの所望の部分にブレークポイントを設定するステップと、与えられたスクリプトの実行に用いる所望の変数を確認の対象として指定するステップと、与えられたスクリプトを指定された態様で実行するステップと、設定された前記ブレークポイントにおいてスクリプトの実行を停止するステップと、指定された変数に関する情報を出力するステップと、を含むことを特徴とする。請求項18、20の発明では、指定したブレークポイントでスクリプトの実行が停止し、1ステップずつ実行したり、次のブレークポイントまで実行するなど所望の態様でスクリプトの実行を自由に制御することができ、また、指定した変数に関する情報が得られるので、複雑なスクリプトについても効果的にデバッグを行うことができる。

【0047】

【発明の実施の形態】次に、本発明の実施の形態（以下「実施形態」という）について、図面を参照して説明する。なお、本発明は、周辺機器を持つコンピュータを、ソフトウェアで制御することによって実現するが一般的と考えられる。この場合、キーボード及びマウスなどの入力装置で情報を入力し、CRT表示装置及びプリンタなどの出力装置で情報を出力できる。また、レジスター、メモリ、外部記憶装置などの記憶装置は、いろいろな形式で、情報を一時的に保持したり永続的に保存できる。そして、CPUは、前記ソフトウェアにしたがって、これらの情報に加工及び判断などの処理を加え、さらに、処理の順序を制御することができる。

【0048】また、コンピュータを制御するソフトウェアは、各請求項及び明細書に記述する処理に対応した命令を組み合わせることによって作成され、作成されたソフトウェアは、コンパイラやインタプリタなどの処理系によって実行されることで、上記のようなハードウェア資源を活用する。

【0049】但し、本発明を実現するための上記のような態様はいろいろ変更することができ、例えば、本発明の装置と外部との間で情報を入出力するには、フロッピーディスクなどの着脱可能な記録媒体やネットワーク接続装置を使用することもできる。さらに、本発明を実現するソフトウェアを記録したCD-ROMのような記録媒体は、それ単独でも、本発明の一態様である。また、本発明の機能の一部をLSIなどの物理的な電子回路で実現することも可能である。

10 【0050】以上のように、コンピュータを使用して本発明を実現する態様はいろいろ変更できるので、以下では、本発明の各機能を実現する仮想的回路ブロックを用いることによって、本発明の実施の形態を説明する。

【0051】〔1. 第1実施形態…モニタ装置〕第1実施形態は、請求項1～14に対応するもので、エージェントが複数のノード上を移動しながら情報処理を行うエージェントシステムをモニタするためのモニタ装置において、システム全体を監視するシステムモニタと、ノードを監視するノードモニタと、エージェントを監視するエージェントモニタと、を備えたものである。

【0052】〔1-1. 構成〕図1は、第1実施形態の構成を示す概念図である。この図に示すように、第1実施形態は、エージェントシステム全体を監視するシステムモニタ111と、ノード112を監視するノードモニタ113、エージェント114を監視するエージェントモニタ115を有する。これらの各モニタは、例えば、ノード上で実行されるプロセスとして立ち上げる。

【0053】このうち、システムモニタ111は、エージェントシステムのネットワークに接続されたどのノード上に設けてもよいが、ノード以外の、エージェントシステムがネットワーク経由で参照可能な任意のホスト（マシン）上で立ち上げることも可能である。こうすることにより、エージェントシステム外からシステムを監視することもできる。

【0054】また、ネーミングシステム119との通信量を削減するためには、システムモニタ111はネーミングサービス119の存在するホスト上に設けることが望ましい。そして、ネーミングサービス（ネームサーバ）119も、ノード以外の、エージェントシステムがネットワーク経由で参照可能な任意のホスト上で立ち上げることが可能である。

【0055】また、ノードモニタ113は、特定の1つのノードを監視するもので、複数のノードを監視するためにはノード毎に同数のノードモニタ113を立ち上げて用いる。

【0056】また、エージェントモニタ115も、特定の1つのエージェントを監視するもので、複数のエージェントに対しては同数立ち上げて用いる。このエージェントモニタは、対応するエージェントが存在するノードのノードモニタ113が、当該ノードモニタ113の一

部として保持する。

【0057】また、これらシステムモニタ111、ノードモニタ113、エージェントモニタ114は、それぞれ対応する表示画面であるシステムモニタウインドウ116、ノードモニタウインドウ117、エージェントモニタウインドウ118を介して、ユーザとの間でデータを入出力したり、指示や操作・各種の指定を受け付ける。そして、これら各モニタ111、113、115は、ユーザが個別に直接立ち上げることもできるし、相互に他のモニタを起動することもできる。

【0058】ここで、ノードモニタ113は、ノードの情報の収集しやすさの点では、監視の対象とするノード上に設けることが望ましく、ノードモニタ113が収集した情報は、表示画面であるノードモニタウインドウ117を通じてユーザに提示され、ユーザからの指示もノードモニタウインドウ117を介して行われる。

【0059】このノードモニタウインドウ117は、遠隔地にあるノードモニタ113とユーザとの通信を行うのが主な役割である。よって、このように遠隔地からノードをモニタする場合の通常の構成例としては、ノード上にノードモニタ113が存在し、このノードをモニタするユーザ側のマシン上でノードモニタウインドウ117が立ち上がっているという形態が考えられる。そして、モニタ用の各ウインドウ116～118は、エージェントシステムがネットワーク経由で参照可能な任意のホスト上で立ち上げることが可能である。

【0060】〔1-2. 作用〕上記のような構成を有する第1実施形態は、次のような作用を有する。ここで、図2は、第1実施形態における処理手順を示すフローチャートである。

〔1-2-1. システムモニタの作用〕まず、ユーザが最初にシステムモニタ111を立ち上げると（ステップ6211）、システムモニタ111は、エージェントシステムの持つネーミングサービス119と通信することによってシステム全体の情報を獲得し、獲得した情報を表示画面であるシステムモニタウインドウ116に表示する（ステップ6212）。システムモニタ111がこのように表示する情報は、具体的には、例えばエージェントシステム内にどのようなノード及びエージェントが存在するかのリストである。

【0061】ユーザは、このリストから所望のノードを指定することによって（ステップ6213）、そのノードを監視するノードモニタ113を起動（立ち上げ）することができる（ステップ6221）。また、同様に、ユーザはリストからエージェントを指定することによって（ステップ6214）、そのエージェントを監視するエージェントモニタ115を立ち上げることができる（ステップ6231）。なお、ノードやエージェントを指定するには、例えば、名称をキーボードからタイプ入力したり、リスト上のノード名やエージェント名をマウ

スでダブルクリックすればよい。

【0062】〔1-2-2. ノードモニタの作用〕ノードモニタ113は立ち上げられると（ステップ6221）、対応するノード112と通信することによってそのノード112に関する情報を獲得し、獲得した情報を表示画面であるノードモニタウインドウ117に表示する（ステップ6222）。ノードモニタ113がこのように表示する情報は、例えば、対応するノード112にどのようなエージェント114が存在するかのリストや、そのノード112に残されているエージェントログファイル（以下単に「ログ」という）のリストなどである。

【0063】ここで、ログとは、特定のエージェントがプランに基づいてどのような振る舞いをしたかを、プランに含まれる命令の1ステップごとに記録した情報である。ユーザは、このようなノードモニタ113の表示画面において、所望のエージェント名をマウスによるダブルクリックで指定することによって（ステップ6223）、そのエージェントを監視するエージェントモニタ115を立ち上げることができる（ステップ6231）。

【0064】また、ユーザは、ノードモニタの表示画面において所望のログを指定することによって（ステップ6224）、そのログに基づいてエージェントの動作を確認することができる。具体的には、そのログに基づいてエージェントが生成され、ログに記録されているとおりの振る舞いをそのエージェントに行なわせたり、その先の動作を継続させたりすることができる。

【0065】〔1-2-3. エージェントモニタの作用〕また、エージェントモニタ115は、対応するエージェントと通信することによってそのエージェントに関する情報を獲得し、獲得した情報を、表示画面であるエージェントモニタウインドウ118に表示する（ステップ6232、6233）。エージェントモニタ115が表示する情報は、例えば、そのエージェントがどのノードにいるか（ステップ6232）や、エージェントが現在プランのどの部分をどのように実行しているかといった実行状態（ステップ6233）などである。

【0066】〔1-2-4. ブレークポイントの設定〕また、エージェントモニタ115は、エージェントの動作を制御する機能を持っており、ユーザは、エージェントモニタ115を用いてエージェントのプランの所望の部分にブレークポイントを設定し（ステップ6234）、そのブレークポイントで実行を停止させたり、停止した位置からプランに含まれる各命令を1ステップずつ実行させたり、次のブレークポイントまで実行させたりすることができる。

【0067】このようなブレークポイントは、どんな条件でプランの実行を停止させるかなどに応じて、いくつもの種類を設定することが可能である。また、ブレーク

ポイントを設ける対象についても、例えばプラン中の特定の行を単位としたり、特定の命令を単位としたり、また、特定のゴールやサブゴールを単位とすることができる。さらに、前記のようなブレークポイントの種類や設定対象を自由に組み合わせることもできる。そして、このように設定されたブレークポイントは、エージェントモニタ115の表示画面であるエージェントモニタウインドウ118上に表示される。

【0068】〔1-2-5. ロギングポイントの設定〕また、ユーザは、エージェントモニタ115を用いて、ブレークポイントと同様に、対応するエージェントのプランのうち所望の部分にロギングポイントを設定することができます。ここで、ロギングポイントとは、エージェントの動作についてログの記録を制御するためのポイントであり、ログの記録を行う開始点や終了点を指定するのに用いる。また、ロギングポイントは、ブレークポイントと同様に様々な条件や対象の単位を組み合わせて設定することができる。例えば、プランのどこからどこを実行しているときにログを記録するかという範囲だけでなく、例えば、実行失敗に対する復旧処理の間だけ記録するなど、ログを記録する条件も指定することができる。

【0069】そして、エージェントのプランが実行されると、エージェントモニタ115は、ロギングポイントで指定した条件や対象について、エージェントのログファイルを記録し、ユーザの求めに応じてエージェントモニタウインドウ118に出力する。さらに、このように得られたログファイルをノードモニタ113上から利用することによって、そのログをとったときと同じ状態で、エージェントを再現し、ログに記録された動作を再現させたり、その先の動作の実行を継続させたりすることができる。

【0070】〔1-2-6. エージェントが移動する場合の処理〕ところで、エージェントモニタ115が対応しているエージェントが別のノードに移動すると(図3)、そのエージェントモニタ115を保持しているノードモニタ113は、当該エージェントモニタ115を破棄し、エージェントの移動先のノードに対応するノードモニタ113内に新たにエージェントモニタ115が生成される。

【0071】また、このようにエージェントモニタ115が移動しても、ユーザに対する表示画面としては同じエージェントモニタウインドウ118を使用する必要がある。このため、エージェントの移動によってエージェントモニタ115が破棄された場合、エージェントモニタウインドウ118は、図3に示すように、同じエージェントに対応するエージェントモニタ115が新たにどのノードで生成されたかをシステムモニタ111に問い合わせる。そして、エージェントモニタウインドウ118は、システムモニタ111からの回答に基づいて、新

たに生成されたエージェントモニタ115との間にデータのリンクを確立し、エージェントモニタ115から送られてくる情報を表示したり、ユーザからの指示をエージェントモニタ115に転送する処理を続ける。

【0072】〔1-3. 効果〕上記のように、第1実施形態では、システム全体、ノード及びエージェントをそれぞれ専用のモニタによって監視できるので、エージェントシステムの全体と各部分の双方について状態を効果的に把握でき、エージェントシステムの管理やデバッグが容易になる。特に、第1実施形態では、各モニタから他の種類のモニタを起動できるので、エージェントシステムの状態を多面的に把握することが容易になる。

【0073】また、第1実施形態では、所望のエージェントのログファイルに基づいてエージェントの動作を再現などによって確認できるので、個々のエージェントの動作を容易に検討することができる。また、第1実施形態では、エージェントがノード間で移動すると、移動したエージェントのエージェントモニタは、移動元ノードでは破棄され、移動先ノードで再度生成されるので、エージェントモニタをエージェントと共に送信する必要がなく、そのような送信に要する所要時間や手数が不要となる。

【0074】また、第1実施形態では、エージェントがどのノードにおいて、プランのどの部分を実行しているかがoutputできるので、所望のエージェントの状態把握が容易になる。また、第1実施形態では、エージェントモニタからエージェントにブレークポイントを設定することができるので、エージェントの状態の監視からデバッグへ容易に移行でき、作業が効率化される。また、第1実施形態では、エージェントのプランにロギングポイントを設定することによって、必要な部分だけのログを容易に記録することができる。

【0075】〔2. 第2実施形態…デバッグ装置〕第2実施形態は、請求項15～21に対応するもので、エージェントシステムをデバッグするためのデバッグ装置である。このデバッグ装置がデバッグの対象とするエージェントシステムは、知識及びアクション定義に基づいてプランナがエージェントのプランを生成し、エージェントは生成されたプランを実行することによって情報処理を行うものである。

【0076】第2実施形態は、このようなエージェントシステムに用いられる知識、アクション定義及びプランナについて、それらを用いたときに生成されるプランのスクリプトが正しいかどうかを検討することによって、それら知識、アクション定義及びプランナをデバッグするものである。なお、エージェントが実行する行動プログラム全体がプランすなわちスクリプトであり、また、プランナは、プランすなわちスクリプトの生成に必要なデータとして知識やアクション定義を参照し、プラン(スクリプト)生成の手続きを行うプログラムである。

【0077】〔2-1. 構成〕まず、図4は、第2実施形態の構成を示す機能ブロック図である。この図に示すように、第2実施形態は、ナレッジデバッガDと、スクリプトトレーサTとを有する。ナレッジデバッガDは、デバッグの対象として指定された知識、アクション定義及びプランナを用いて、与えられたゴールを達成するためのスクリプトS1を生成する手段である。また、スクリプトトレーサTは、与えられたスクリプトの挙動を調べる手段である。

【0078】このうちナレッジデバッガDは、デバッグの対象を設定するための第1の設定手段D1と、設定されたデバッグの対象に基づいてスクリプトを生成する生成手段D2と、生成されたスクリプトの表示を制御する表示制御手段D3と、を有する。

【0079】また、スクリプトトレーサTは、第2の設定手段T1と、実行制御手段T2と、出力制御手段T3と、を有する。このうち第2の設定手段T1は、与えられたスクリプトについてブレークポイントを設定し及び監視したい変数を指定する手段である。また、実行制御手段T2は、与えられたスクリプトを指定された態様で実行する手段であり、このような実行の態様としては例えば1ステップずつ実行するのか全部まとめて実行するのかなどが選択できる。また、実行制御手段T2は、ブレークポイントにおいてスクリプトの実行を停止するように構成されている。また、出力制御手段T3は、スクリプトの実行で得られた情報、例えば指定された変数の値などを出力する手段である。

【0080】〔2-2. 作用〕上記のような第2実施形態を用いたデバッグの手順を図5に示す。なお、第2実施形態では、ユーザは、GUI（グラフィカルユーザインターフェース）を用いて各種設定を行なったり、指示を与えて、情報を参照することができる。

【0081】〔2-2-1. デバッグ対象の指定〕まず、ユーザは、デバッグ対象とする知識、アクション及びプランナに対して従来のエディタなどを用いて必要な編集や修正を加える（ステップ7201）。続いて、ナレッジデバッガDの第1の設定手段D1を用いて、デバッグ対象とする知識、アクション及びプランナを指定する（ステップ7202）。具体的には、例えば、知識、アクション定義、プランナを複数ある候補から選択して利用できる場合、入出力用の表示画面（ダイアログ）にこれらの候補を表示させ、その中からデバッグの対象とするアクション定義やプランナなどを選択する。また、必要に応じて目的とする情報のファイルを指定し、ファイルから情報を読み込んでデバッグすることもできる。選択されたこれらの情報は、所定のテキスト表示欄（テキストボックス）に表示される。

【0082】〔2-2-2. スクリプトの生成〕デバッグの対象とする情報を上記のように選択した後、ユーザは、ダイアログから所望のゴールを入力して設定する。

統いて、所定の操作、例えば表示画面上にグラフィック表示されているGetScriptボタンをマウスでクリックすることによって生成手段D2を起動する。起動された生成手段D2は、選択されている知識、アクション定義及びプランナを用いて、与えられたゴールを達成するためのスクリプトを生成し（ステップ7203）、生成されたこのスクリプトが所定の表示用ウインドウに出力される。

【0083】〔2-2-3. サブゴールの抽出〕このように生成されたスクリプトをどのように表示するかは、表示制御手段D3が制御する。すなわち、まず、表示制御手段D3は、生成されたスクリプトに含まれるサブゴールを抽出し（ステップ7204）、スクリプト自体とは別の表示用ウインドウにサブゴールリストとして表示される。このようなサブゴールは、スクリプト内では決まった命令語、例えばsubgoalやnewgoalなどを用いて表されるので、サブゴールを抽出するには、これらの命令語をスクリプト内から検索すればよい。このように抽出され表示されたサブゴールのなかから、ユーザが所望のサブゴールGを選択することによって（図4）、そのサブゴールGを達成するための部分的なスクリプトS2だけを生成させて出力させることもできる。

【0084】〔2-2-4. スクリプトの階層的な表示〕さらに、表示制御手段D3は、ゴールとサブゴール及び前記ゴールに対応するスクリプトと前記サブゴールに対応する部分的なスクリプトを、階層関係に基づいて表示することができる（ステップ7205）。すなわち、最終的なゴールとサブゴールの関係は、最終的なゴールが上位であり、最終的なサブゴールを達成するための各サブゴールが下位という階層的な関係にある。このため、下位のサブゴール及びサブゴールに対応する部分的なスクリプトは、スクリプト全体との関係で、階層的に管理され、表示される。

【0085】例えば、あるサブゴールを具体的なスクリプトに展開して表示させた後、表示を上位階層に戻すUpボタンをマウスでクリックすることによって、展開したサブゴールを含んでいた上位のスクリプト全体や、この全体的なスクリプトに対応する最終的なゴールの出力画面に戻ることもできる。同様に、サブゴールのスクリプトがさらに下位のサブゴールを含んでいる場合は、階層的に順次下位のスクリプトを展開して表示せたり、逆に、上位の表示画面に順次戻ることもできる。

【0086】ここで、スクリプト中のサブゴールに対応する部分的なスクリプトを表示させ、表示されたスクリプト中のサブゴールに基づいてさらに下位の情報を表示していく状態を図6の概念図に示す。このようにスクリプトを表示した結果、バグを発見した場合は（ステップ7206）デバッグ対象の編集／修正（ステップ7201）に戻って作業をやり直せばよい。

【0087】〔2-2-5. スクリプトのトレース〕ま

た、上記のように得られた所望のスクリプトをスクリプトトレーサTに渡すことによって、スクリプトをトレースすることができる。ユーザは、スクリプトをトレースしようとするときは、トレースの対象とするスクリプトを例えば反転表示によって選択した状態で、表示画面上のTraceボタンをマウスでクリックする。すると、スクリプトトレーサTが起動され(ステップ7207)、スクリプトトレーサTの入出力用ウインドウが現われ、渡されたスクリプトがテキスト表示された状態となる。

【0088】〔2-2-6. ブレークポイントと変数ウォッチ式の設定〕このテキスト表示画面上で、ユーザは、第2の設定手段T1を用いて、所望の位置に何箇でもブレークポイントを設定することができる(ステップ7208)。また、ユーザは、第2の設定手段T1を用いて、スクリプト内で使われている変数のうち、所望の変数を変数ウォッチ式で指定する(ステップ7208)。ここで、変数ウォッチ式は、スクリプトを実行する際にどの変数を監視し、その変数にどのような変化があったときに、その変化をどのような形式でユーザに提示するかを指定する式である。

【0089】このような変数ウォッチ式によって、例えば、ある2つの変数の値が食い違ったときに警告を表示したり、ある変数のうち所定の範囲のビットを1と0の並びで表示するといった指定を自由に行うことができる。変数ウォッチ式によって指定されたこのような表示は、出力制御手段T3によって実現される。

【0090】〔2-2-7. スクリプトの実行〕以上のようにブレークポイント及び変数ウォッチ式を設定したスクリプトは、実行制御手段T2によって実行される(ステップ7209)。この際、ユーザは、実行制御手段T2を利用して、スクリプトを1ステップずつ実行することもできるし、各ステップを連続して実行することもできる。そして、このようなスクリプトの実行に際して、出力制御手段T3は、変数ウォッチ式で指定した変数の内容を順次表示及び記録する。

【0091】また、各ステップを連続して実行する場合も、実行制御手段T2は、設定したブレークポイントで実行を停止するので(ステップ7209)、ユーザは、その時点で各変数の値がどうなっているか確認することができる(ステップ7210)。また、1ステップずつ実行する場合は、スクリプトのどの部分がどんな順序で実行され、各変数の値がどのように変化していくかを詳細に検討することができる。この場合も、スクリプトが実行される様子や変数の内容を確認した結果、バグが発見された場合は、デバッグ対象の編集／修正(ステップ7201)から作業をやり直せばよい。

【0092】〔2-3. 効果〕以上のように、第2実施形態では、指定されたデバッグ対象を用いて、与えられたゴールを達成するためのスクリプトが生成され出力されるので、ユーザは、所望のデバッグ対象について意図

した通りのスクリプトが生成されているかどうかを容易に確認することができる。

【0093】また、第2実施形態では、最終的なゴールに対応するスクリプト全体を表示して正しいかどうかを判断するだけでなく、所望のサブゴールに対応する部分的なスクリプトを表示して詳細に検討できるので、スクリプトの理解が容易になる。また、第2実施形態では、ゴールとサブゴール及びゴールに対応するスクリプトとサブゴールに対応する部分的なスクリプトについて、上位階層の表示と下位階層の表示を切り替えるなど階層関係に基づいて表示することによって、所望の範囲を容易に確認することができる。

【0094】また、第2実施形態では、指定したブレークポイントでスクリプトの実行が停止し、1ステップずつ実行したり、次のブレークポイントまで実行するなど所望の態様でスクリプトの実行を自由に制御することができ、また、指定した変数にの内容を調べることができますので、複雑なスクリプトについても効果的にデバッグを行うことができる。

20 【0095】〔3. 他の実施の形態〕なお、本発明は上記実施の形態に限定されるものではなく、次に例示するような他の実施形態をも含むものである。例えば、本発明において、ネットワークの規模、形式、ノード数などは自由であり、エージェントを用いて行う情報処理の種類や、アクション定義、知識、プランなどを記述する言語の形式も自由に選択しうる。

【0096】また、第1実施形態において、システムモニタ、ノードモニタ、エージェントモニタの具体的な構成や情報を表示する形式は自由であり、これら各モニタや各モニタの表示画面を一体に構成することもできる。また、各モニタから他のモニタを起動する機能や、ログファイルを取り扱う機能や、エージェントモニタからブレークポイントやロギングポイントを設定できる機能は、必ずしも必要ではない。また、エージェントが他のノードへ移動する場合、必ずしもエージェントモニタを破棄する必要はなく、エージェントと共に移動先に転送してもよい。

【0097】また、第2実施形態において、デバッグの対象として選択できるのは知識、アクション定義及びプランに限定する必要はなく、システム構成に合わせて他の所望の構成要素を加えることができる。また、デバッグの対象を全てユーザが選択する必要はなく、選択しなかった対象についてはシステム上の標準構成(デフォルト)が選択されるようにすることもできる。また、生成されたスクリプトを表示する場合、表示は必ずしも階層的にする必要はなく、また、生成したスクリプトにブレークポイントを設定したり、指定した態様で実行する手段も必須ではない。

【0098】

【発明の効果】以上説明したように、本発明によれば、

全体、ノード、エージェントといった単位でそれぞれモニタで監視することによってエージェントシステムの振る舞いを把握できるので、エージェントシステムの管理と開発とが効果的に支援される。

【図面の簡単な説明】

【図1】本発明の第1実施形態の構成を示す機能ブロック図。

【図2】本発明の第1実施形態における処理手順を示すフローチャート。

【図3】本発明の第1実施形態において、エージェントがノード間で移動した場合の処理を示す概念図。

【図4】本発明の第2実施形態の構成を示す機能ブロック図。

【図5】本発明の第2実施形態における処理手順を示すフローチャート。

【図6】本発明の第2実施形態において、階層的な表示を説明するための図。

【図7】従来のエージェントシステムの構成例を示すフローチャート。

【図8】プランニングを行う従来のエージェントシステムの構成例を示すフローチャート。

【図9】従来のエージェントシステムにおける処理手順を示すフローチャート。

【図10】エージェントシステムにおいて、生成途中のプランの例を示す概念図。

【図11】エージェントシステムにおいて、完成したプランの例を示す概念図。

【図12】エージェントシステムにおけるプランニングの手順を示すフローチャート。

【図13】エージェントシステムにおいて、エージェントがノード間で移動する手順を示すフローチャート。

【図14】エージェントシステムにおけるエージェントのライフサイクルを示す概念図。

【図15】エージェントシステムにおいてノード上に複数のファーム（フィールド）が存在する状態を示す概念図。

【符号の説明】

1 … ローカル情報記憶手段

2 … 更新手段

3 … エージェント情報記憶手段

4 … 入出力手段

5 … プラン生成手段

6 … プラン実行手段

7 … エージェント管理部

L … ローカルマシン

R … リモートマシン

N … ネットワーク

10 1 1 1 … システムモニタ

1 1 2 … ノード

1 1 3 … ノードモニタ

1 1 4 … エージェント

1 1 5 … エージェントモニタ

1 1 6 … システムモニタウインドウ

1 1 7 … ノードモニタウインドウ

1 1 8 … エージェントモニタウインドウ

1 1 9 … ネーミングサービス

D … ナレッジデッパガ

20 D 1 … 対象設定手段

D 2 … 生成手段

D 3 … 表示制御手段

T … スクリプトトレーサ

T 1 … 第2の設定手段

T 2 … 実行制御手段

T 3 … 出力制御手段

S 1, S 2 … スクリプト

C … 条件

P … 動作

30 S … 状態

8 0 0 N … ネットワーク

8 0 0 L … ローカルノード

8 0 0 R … リモートノード

8 0 1 … エージェント情報記憶手段

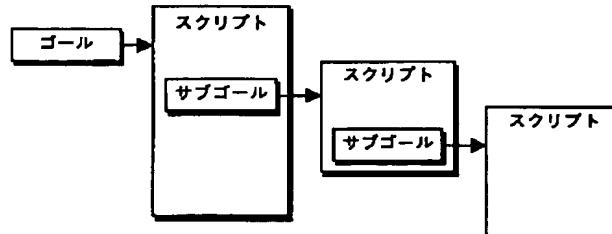
8 0 2 … 解釈実行手段

8 0 3 … 入出力手段

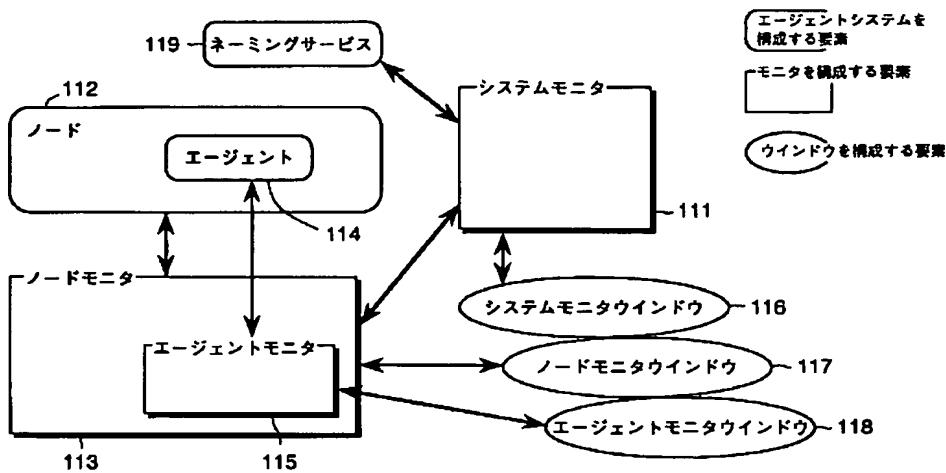
8 0 4 … エージェント管理手段

S T E P … 手順の各ステップ

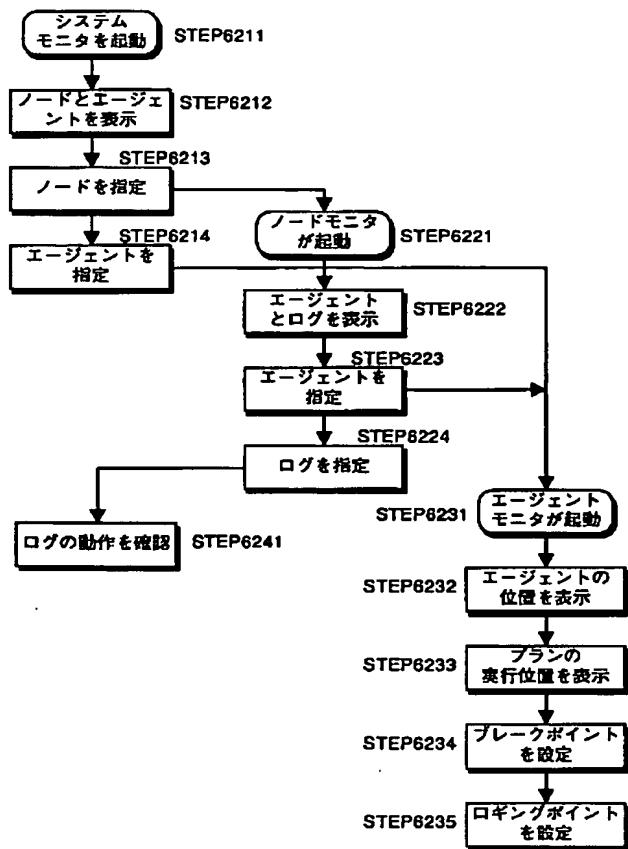
【図6】



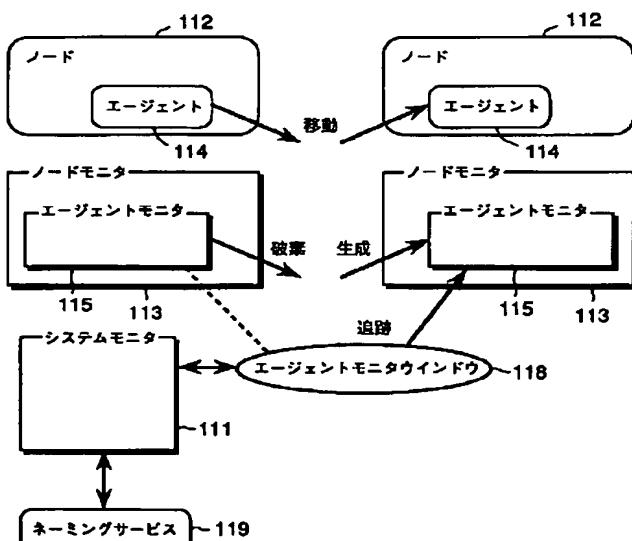
【図1】



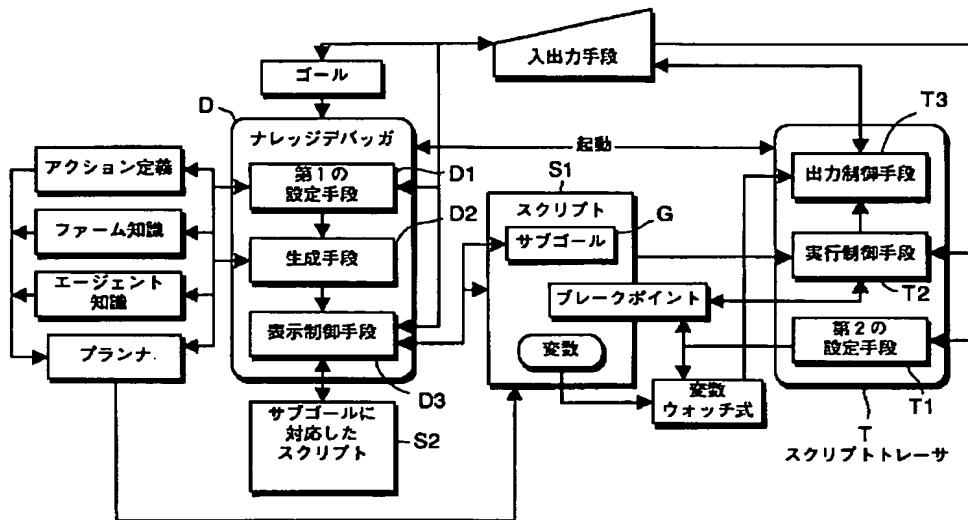
【図2】



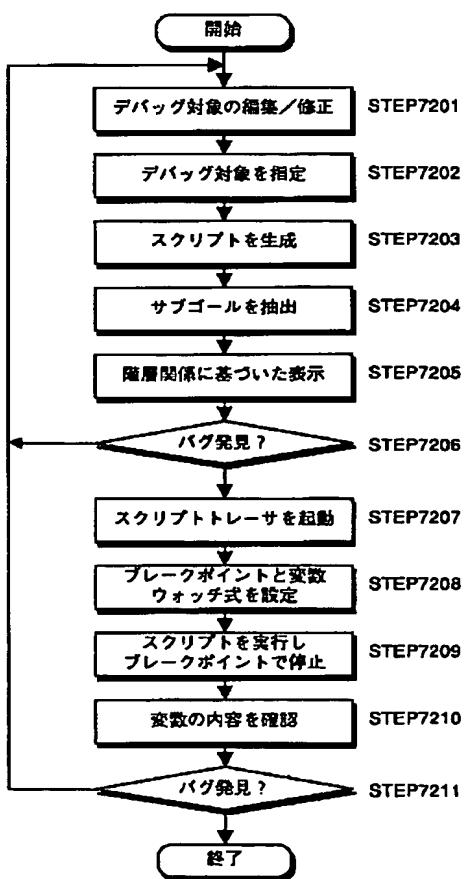
【図3】



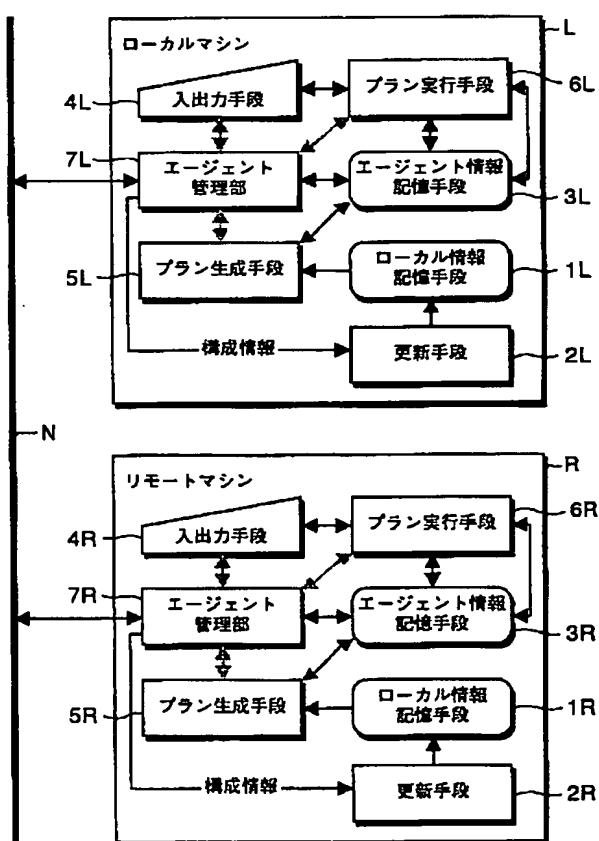
[図4]



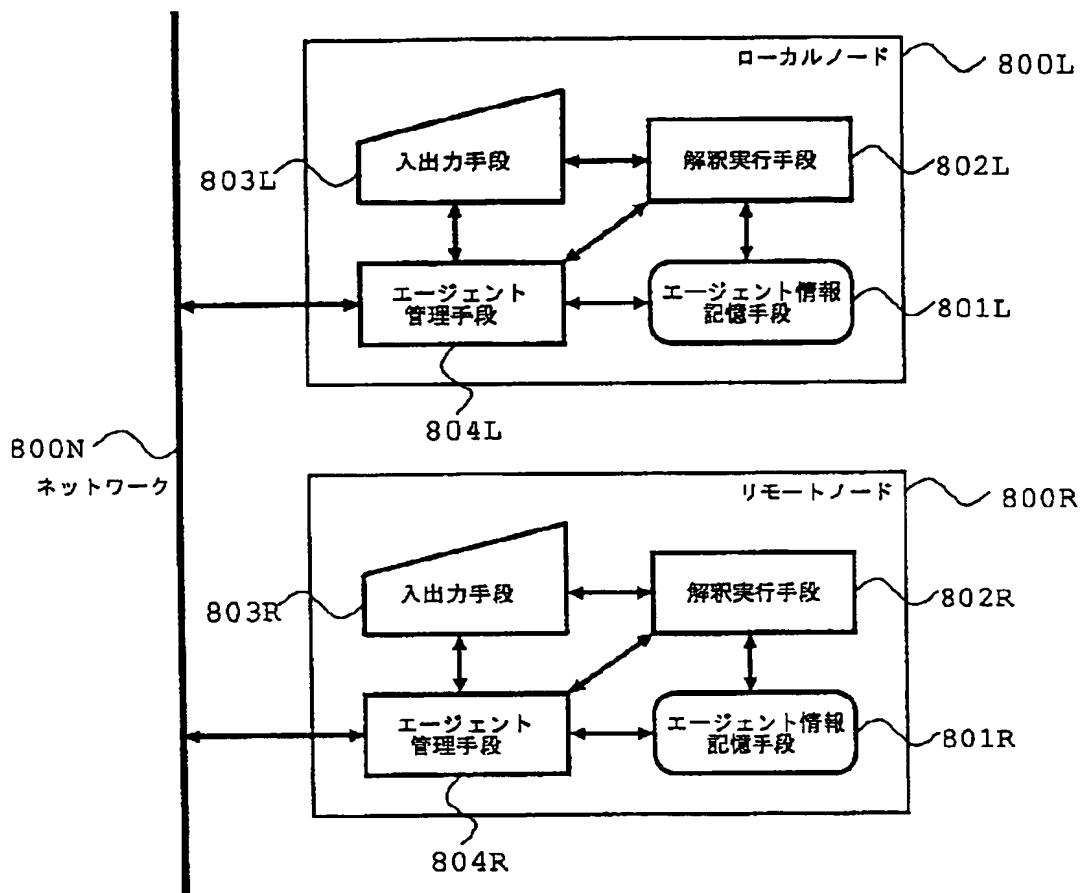
【図5】



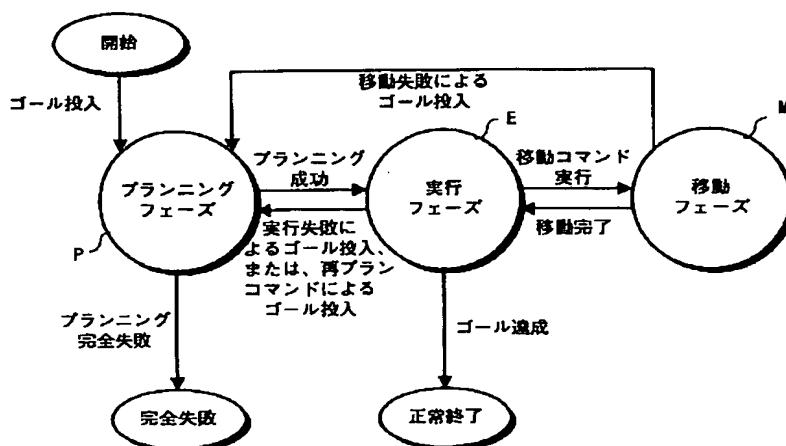
〔图8〕



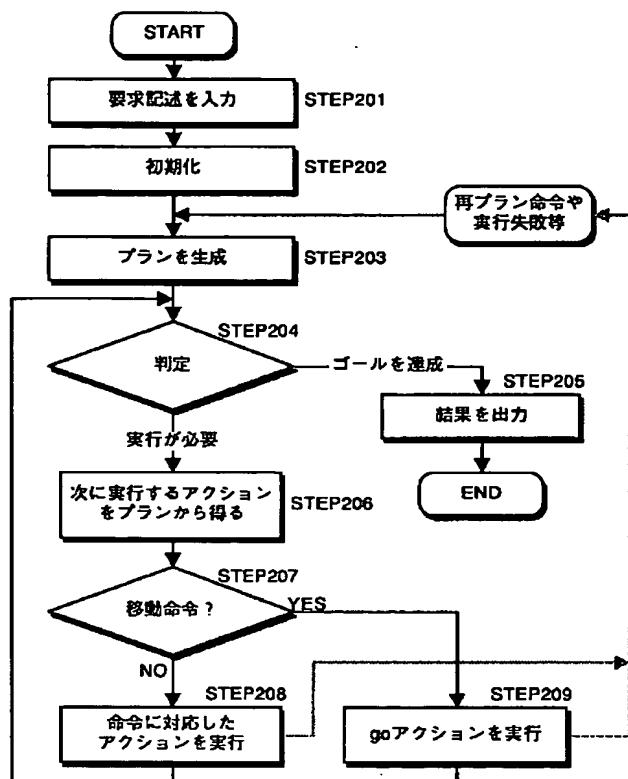
【図7】



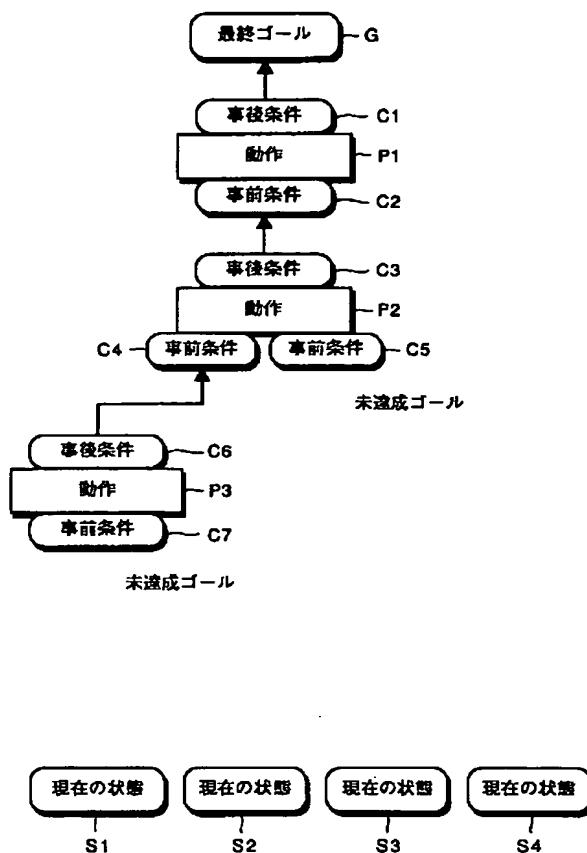
【図14】



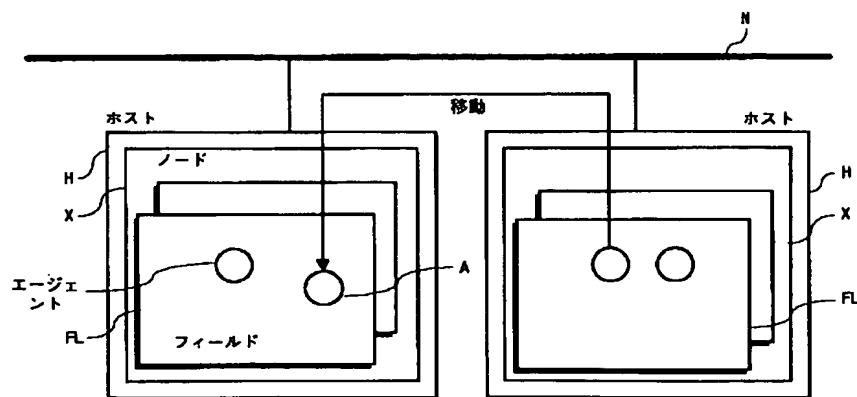
[图 9]



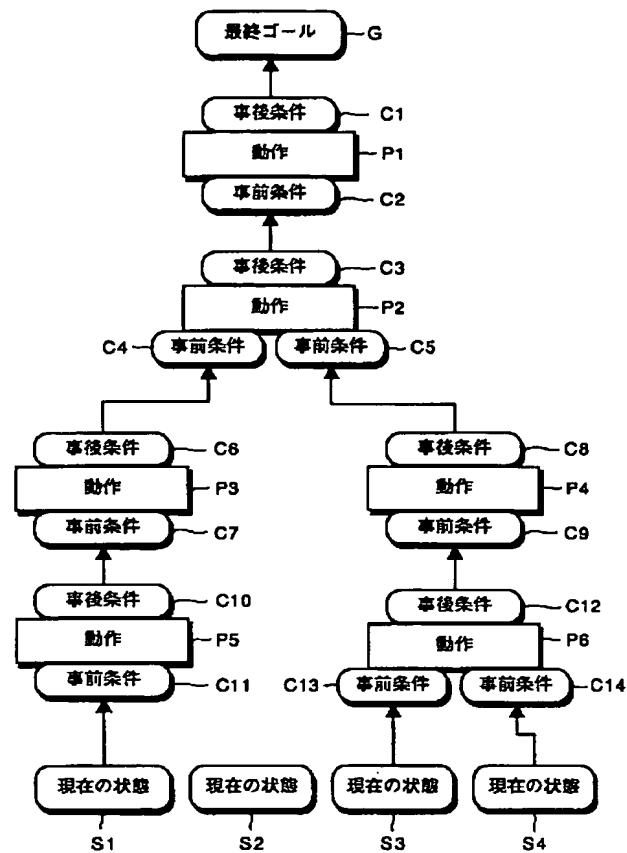
[図10]



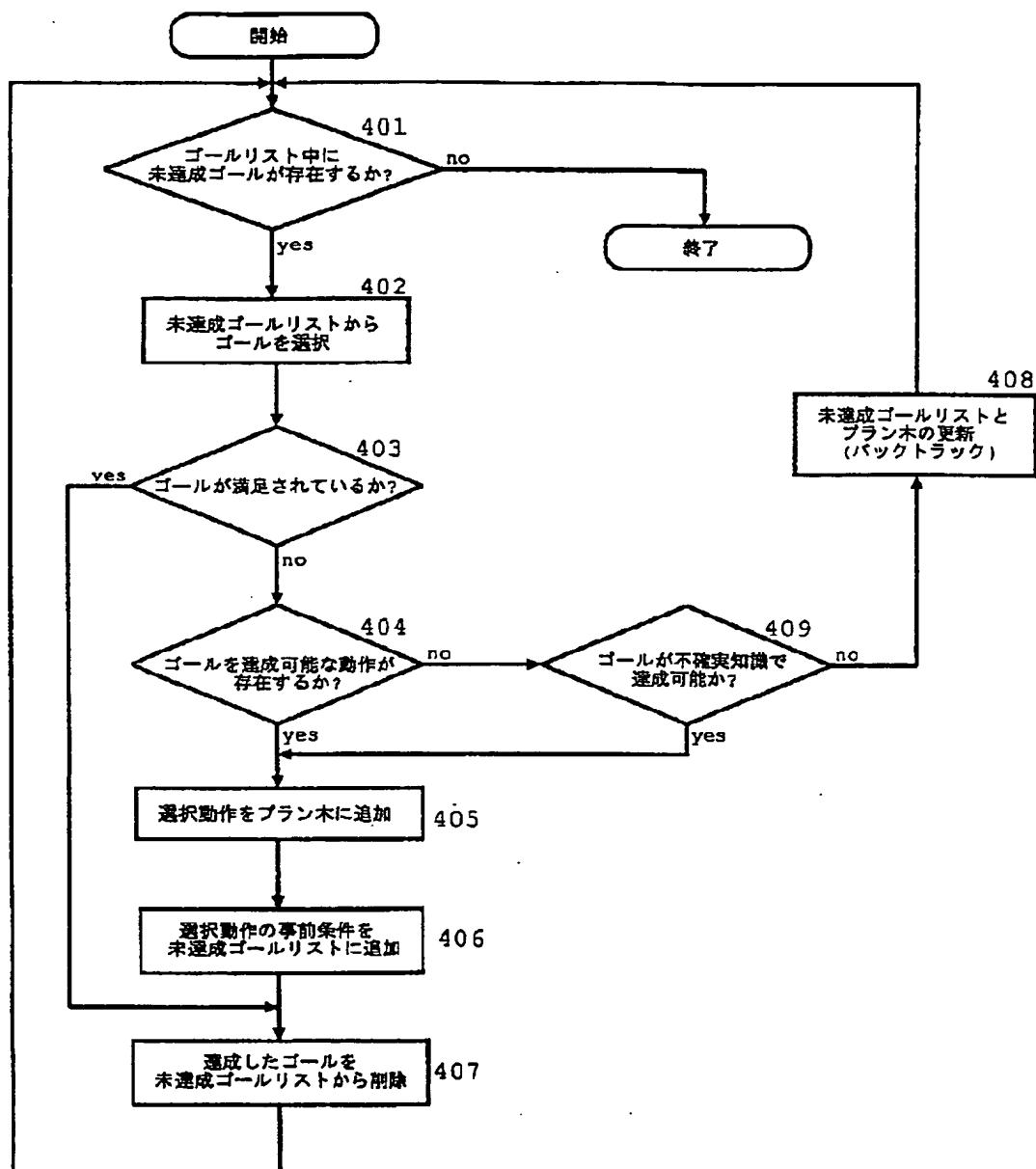
【図15】



【図11】



【図12】



【図13】

